

Creating and Modifying a Scatter Graphics Method

Â

[Contents](#) [Previous](#) [Next](#)

Goal: Guide you through creating and setting scatter graphics method attributes.

Before running the tutorial below, type "python" or "cdat" at the command line.Â You will see the python prompt appear (i.e., ">>>"). You can now enter the command lines below.

You can [view](#) or [download](#) the full source code. To run the source code at the command line, type: "python scatter_file.py".

```
# Import the modules needed for the tutorial
# cdms - Climate Data Management system accesses gridded data.
# vcs - Visualization and control System 1D and 2D plotting routines.
# cdutil - Climate utilitizes that contains miscellaneous routines for
#           manipulating variables.
# time - This module provides various functions to manipulate time values.
# os - Operation System routines for Mac, DOS, NT, or Posix depending on
#       the system you're on.
# sys - This module provides access to some objects used or maintained by
#       the interpreter and to functions that interact strongly with the interpreter.
import vcs, cdms, cdutil, time, os, sys

# Open data file:
filepath = os.path.join(sys.prefix, 'sample_data/clt.nc')
cdmsfile = cdms.open( filepath )

# Extract two 3 dimensional data sets and get a subset of the time dimension
data1 = cdmsfile('u', longitude=(-180, -48.75), latitude = (10., 70.43))
data2 = cdmsfile('v', longitude=(-180, -48.75), latitude = (10., 70.43))

# Set the longitude and latitude axes of the "v" variable
# to that of the "u" variable.
data2.setAxis(2, data1.getAxis(2))
data2.setAxis(3, data1.getAxis(3))

# Initial VCS:
v = vcs.init()

# Show the list of persistent scatter graphics methods.
v.show( 'scatter' )

*****Scatter Names List*****
( 1):          ASD          default        quick
*****End Scatter Names List*****
```

Get a scatter graphics method object and plot:

```
# Assign the variable "sf_asd" to the persistent 'ASD' scatter graphics methods.
sf_asd = v.getscatter( 'ASD' )

# Plot the data using the above scatter graphics method.
v.plot( data1, data2, sf_asd )
```

```

# List the 'ASD' scatter graphics methods attributes.
sf_asd.list()

-----Scatter (GSp) member (attribute) listings -----
Canvas Mode = 1
graphics method = GSp
name = ASD
projection = linear
xticlabels1 = *
xticlabels2 = *
xmtics1 =
xmtics2 =
yticlabels1 = *
yticlabels2 = *
ymtics1 =
ymtics2 =
datawc_x1 = 1.00000002004e+20
datawc_y1 = 1.00000002004e+20
datawc_x2 = 1.00000002004e+20
datawc_y2 = 1.00000002004e+20
datawc_timeunits = days since 2000
datawc_calendar = 135441
xaxisconvert = linear
yaxisconvert = linear
marker = None
markercolor = None
markersize = None

# Change the scatter attributes
sf_asd.marker = 5.0
sf_asd.markercolor = 243
sf_asd.markersize = 15

# Create a persistent scatter graphics methods from an existing scatter graphics method.
sf_new = v.createscatter( 'new', 'ASD' ) # create new from ASD
sf_new2 = v.createscatter( 'new2', 'ASD' )# create new2 from ASD
sf_new.list()                         # list its attributes
v.show( 'scatter' )                   # show vector list with new and new2
v.removeobject( sf_new )              # remove new from vector list
v.show( 'scatter' )                   # show vector list without new

```

Â

[Contents](#) [Previous](#) [Next](#)